

Wright State University

CORE Scholar

Kno.e.sis Publications

The Ohio Center of Excellence in Knowledge-
Enabled Computing (Kno.e.sis)

2008

The 4 X 4 Semantic Model: Exploiting Data, Functional, Non-Functional and Execution Semantics across Business Process, Workflow, Partner Services and Middleware Services Tiers

Amit P. Sheth

Wright State University - Main Campus, amit@sc.edu

Karthik Gomadam

Follow this and additional works at: <https://corescholar.libraries.wright.edu/knoesis>



Part of the [Bioinformatics Commons](#), [Communication Technology and New Media Commons](#), [Databases and Information Systems Commons](#), [OS and Networks Commons](#), and the [Science and Technology Studies Commons](#)

Repository Citation

Sheth, A. P., & Gomadam, K. (2008). The 4 X 4 Semantic Model: Exploiting Data, Functional, Non-Functional and Execution Semantics across Business Process, Workflow, Partner Services and Middleware Services Tiers. *Lecture Notes in Business Information Processing*, 12, 16-26.
<https://corescholar.libraries.wright.edu/knoesis/57>

This Conference Proceeding is brought to you for free and open access by the The Ohio Center of Excellence in Knowledge-Enabled Computing (Kno.e.sis) at CORE Scholar. It has been accepted for inclusion in Kno.e.sis Publications by an authorized administrator of CORE Scholar. For more information, please contact library-corescholar@wright.edu.

THE 4 X 4 SEMANTIC MODEL:

Exploiting Data, Functional, Non-Functional and Execution Semantics Across Business Process, Workflow, Partner Services and Middleware Services Tiers

Amit P.Sheth, Karthik Gomadam

kno.e.sis center, Department of Computer Science and Engineering, Wright State University, Dayton, OH.
{*amit.sheth, gomadam-rajagopal.2*}@wright.edu

<http://knoesis.wright.edu>

Keywords: 4 x 4 Semantic Model, Business Process Tier, Workflow Enactment Tier, Partner Services Tier, Middleware Services Tier, Data Semantics, Function Semantics, Nonfunctional Semantics, Execution Semantics, SAWSDL, SA-REST, Semantic Templates, Micro-formatted Semantic Templates, Smashups

Abstract: Business processes in the global environment increasingly encompass multiple partners and complex, rapidly changing requirements. In this context it is critical that strategic business objectives align with and map accurately to systems that support flexible and dynamic business processes. To support the demanding requirements of global business processes, we propose a comprehensive, unifying 4 X 4 Semantic Model that uses Semantic Templates to link four tiers of implementation with four types of semantics. The four tiers are the Business Process Tier, the Workflow Enactment Tier, the Partner Services Tier, and the Middleware Services Tier. The four types of semantics are Data Semantics, Function Semantics, Nonfunctional Semantics, and Execution Semantics. Our model encompasses services architectures that include enterprise class WSDL-based Web services as well as the lightweight but broadly used REST-based services.

1 Introduction

Globalization and outsourcing provide businesses with new means to cost-effective solutions. At the same time, however, they greatly increase the complexity of managing business processes. This complexity can be attributed largely to the dynamic and distributed nature of the business environment ushered in by globalization. Businesses find themselves involved in complex inter- and intra-organizational transactions, when they make a decision such as outsourcing. In this globalized environment, there is a need for organizations to recognize a variety of factors (such as the socio-economic climate of the partners, the shipping resources and capabilities etc.), which can have an impact on its business objectives. Lack of awareness to changes related to partners can prove catastrophic for organizations, as demonstrated by how Ericsson lost significant market share and incurred negative business transformation through its inability to adapt to events (Sheffi, 2005). Another dimension brought forth by globalization is the manifold increase in the number of options that organiza-

tions have when it comes to choosing partners. While on one hand, the aggressive and competitive nature of the global market place helps organizations to reduce costs, the challenges on the other hand lie in selecting those partners that maximize the profitability and minimize the risk. To address these challenges, organizations need to create and enact business process that are very flexible and at the same time, also be able to adapt to the various factors mentioned earlier. Organizations are increasingly trying to leverage the XML driven and loosely couple nature of Service Oriented Architectures (SOA) to realize these objectives.

While the growing adoption of approaches driven by SOA has made it possible to create processes that span multiple partners, the lack of a unifying model to integrate the business processes with the corresponding SOA enactment (i.e., workflow instantiations/enactment) makes it difficult for businesses to create and manage workflows that are consistent with their business objectives. Poor enactment can significantly impede the realization of the objectives of a business process. For example, a cost constraint not modeled correctly at the time of enactment can re-

sult in a suboptimal execution. Further, to translate the dynamism that is envisioned at the level of business processes, one would have to be able to create partner-level requirements from the business process level requirements. Lack of a unifying model makes it hard to (1) create partner-level requirements that are consistent with those of the business process, (2) verify the correctness of the enactment with respect to the business process modeling (3) select and configure the partners at run time, and (4) identify and adapt in an efficiently to the various events that affect the optimality of the business process.

In this paper we show how semantic Web technologies can help create a comprehensive model, called 4×4 Model that extends from business process modeling issues to very detailed descriptions of implementation and enactment. The tiers of our model are: (1) Business Process Tier, (2) Workflow Enactment Tier, (3) Partner Services Tier, and (4) Middleware Services Tier. Adding semantics enhances the process and partner level objectives and constraints captured at each of these tiers. We revisit the four types of semantics for the Web service life cycle initially defined in (Sheth, 2003) and (Sivashanmugam et al., 2003), in light of our model. Adding semantics enhances the

1. Data and functional descriptions at the different tiers using SAWSDL (SAWSDL, 2007)
2. Non-functional descriptions using policy frameworks such as semantically annotated WS-Agreement (Oldham et al., 2006) and
3. Descriptions of lightweight/REST services using microformats and SA-REST (Semantic Annotation of REST, being developed by our group on the lines of WSDL-S (Akkiraju et al., 2004) and SAWSDL).

The rest of the paper is organized as follows: We present a scenario based on real-world use cases that outlines the need for and importance of a unifying process model. The 4×4 Model is presented in section 3. In section 4 we realize the 4×4 model using SAWSDL, enhanced policy descriptions, and SA-REST. In section 5 we present a model for the business process illustrated in the motivating scenario using the 4×4 Model. We present our conclusions in section 6.

2 Motivating Scenario

In this section, we outline a scenario that motivates the need for creating a unifying model for business processes. Our scenario is influenced by the

white paper on Xbox production management published by Microsoft (Microsoft Corporation, 2006).

Microsofts white paper on Xbox production management outlines their adoption of Web services based supply chain management. To optimize production and maintain the production schedule for Xbox, Microsoft has implemented a RosettaNet-driven process framework. Microsoft sends purchase orders to various suppliers that conform to the RosettaNet Request Purchase Order. Once the product is shipped, Microsoft receives shipment notifications conforming to RosettaNet standards. On receiving the shipment, the production units notify the central supply chain management system. Further, the suppliers give Microsoft a view into their production unit so Microsoft can adapt to changes in the production schedule. This example clearly shows the growing acceptance of automated business processes and emphasizes the importance of organization's need to capture such factors as suppliers change in production schedule into their business process models. The scenario below models the business process of a manufacturer similar to Microsoft, using the four tiers.

Consider the procurement process of a gaming hardware manufacturer, which will first involve identifying the types and quantities of parts to be procured. The constraints are the forecasted demand along with the budgetary allocations. The next step is to identify the suppliers for the various parts. During this step, constraints related to the cost of the parts, the relationship between the manufacturer and the suppliers, and the qualitative guarantees offered by the suppliers need to be addressed. In the event of various failures (both system failures such as network unavailability and business-level failures such as delay in receiving shipments), the process needs to adapt while maintaining the optimality requirements. The business analysts will model the requirements to place a purchase order for various parts along with certain constraints such as cost, choice of partners, and supply time. This requirement will then be enacted as a workflow. During enactment, the requirements of the partners of this workflow who will perform various tasks will be modeled. For the manufacturer to meet its business-level objectives, the requirements that are captured at the level of each partner must be consistent with the process-level requirements. The manufacturer must be able to identify events that will affect its objectives and must create schemes toward adapting to those events. The SOA infrastructure that executes this workflow must be capable of understanding these requirements, select partners that meet the requirements, and configure the process, and execute adaptation rules when various events occur.

This scenario illustrates the need for a unifying model that will allow the business analysts to describe the requirements at the process level and the software experts to capture the requirements at the enactment level for each of the partners. In the next section we present the 4×4 Model that integrates the different tiers of the business process.

3 The 4×4 Model

Here we detail the four-tiered approach to integrating objectives and constraints captured in the business process model, on the one hand, with the objectives of the workflow enactments and services descriptions of the workflow partners, on the other. We will describe each of the four tiers in terms of the abstraction and interaction of requirements.

3.1 A four-tiered approach to business process modeling

Based on abstraction and interaction, we categorize business process modeling into three tiers.

1. The Business Process Tier supports the abstract specification that captures the functional and non-functional requirements of the process. In the example of the manufacturer presented in section 2, the Business Process Tier would capture the creation of a purchase order as a functional requirement. The constraints on the security protocols to be adhered to during enactment, the cost and time requirements related to the various parts being ordered, and the domains related to the process would be captured as non-functional requirements.
2. The Workflow Enactment Tier supports an executable, fine-grained model of the abstract process specification. In this tier, the process-level functional requirements are broken down into tasks and the non-functional requirements are captured at the level of each partner who will be executing these tasks. In our example discussed in section 2, the Workflow Enactment Tier would model the cost and time constraints, the part to be ordered, and the security protocols to be adhered to for each partner. In addition, the actual workflow that will realize the purchase order will be created at this tier.
3. The Partner Services Tier consists of the partners who interact with the workflow. The services tier has the service descriptions along with the non-functional requirements and guarantees for each service. The suppliers who wish to partner the gaming manufacturer in section 2 would belong to the Partner Services Tier.
4. The Middleware Services Tier is responsible for providing adaptation and execution services to the other three tiers. The middleware components responsible for partner selection, process configuration and adaptation are parts of the Middleware Services Tier.

Having defined the different tiers of the business process, we now proceed to a brief discussion about the four types of semantics. Understanding the four types of semantics would help us to define the 4×4 Model that integrates the four types of semantics with the four tiers of the business process.

3.2 Four Types of Semantics

We revisit our earlier work ((Sheth, 2003) and (Sivashanmugam et al., 2003)) in which we defined four types of semantics to capture the entire life cycle of a service. The definition is based on the different aspects that go into modeling, composing, and executing a service.

1. Data semantics capture the semantics of the data in a service, including the semantics of the inputs and output of every operation in a service. One way to model data semantics is by annotating the various elements in the input and output messages of a service.
2. The functional capabilities of a service are captured using functional semantics. Functional semantics can model the functional capabilities of a service at the level of each operation or at the level of a service interface. When used in modeling requirements, the functional semantics capture the functional requirement of a request. Adding annotations to service operations is a way to capture the functional semantics.
3. A service, like any other software component, has both functional and non-functional aspects. The non-functional aspects of a service are represented using non-functional (also called QoS (Cardoso et al., 2004), (Verma, 2006)) semantics. This includes standard policy aspects such as transactions, reliable messaging, and security. In addition to these, business-level constraints such as supply time and cost can also be modeled using non-functional semantics. Just as in the functional semantics, non-functional semantics can capture the various requirements and guarantees at the level of operations, interfaces, and services. They

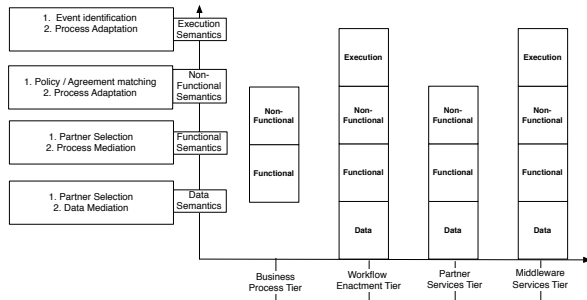


Figure 1: Illustrating the various tiers to model, enact and execute Business Processes

can also be used to capture the non-functional requirements of a service request.

4. The exceptions that may happen during service execution and the strategies and techniques for adapting to them are described using execution semantics. These exceptions include system failures, such as service fault or network unavailability, as well as business-level failures, such as delays in receiving goods or changes in offer price. The idea of non-functional semantics originated as task skeletons in (Krishnakumar and Sheth, 1995) for workflows, and the concepts therein are mapped to suit the services context.

Having defined the four tiers of the business process and the types of semantics, we now present the principle of the 4×4 Model.

3.3 The 4×4 Model

The 4×4 Model integrates the four types of semantics with the four tiers of the business process model. The principle of the 4×4 Model is to capture the explicit relevant semantics at each tier of the business process. This is illustrated in figure 1.

1. The business process tier captures the functional and non-functional requirements of the process at a very abstract level. In the business process tier, the relevant semantics that need to be captured are the functional and the non-functional semantics. If adaptation is modeled at this level, then execution semantics must be captured.
2. The Workflow Enactment Tier needs to capture the requirements for each partner. The data and the control flow to fulfill the functional requirements of the business process are modeled at this tier. The partners to perform various tasks are selected and the process is configured. Further adaptation strategies are modeled and implemented in this tier. To meet these requirements,

all four types of semantics need to be modeled at this tier.

3. The partner services tier interacts with the workflow enactment tier. Addressing data and functional heterogeneities is critical for facilitating interactions. In addition to this requirement, during service selection the non-functional guarantees of a service must meet the requirements captured in the workflow enactment tier. The semantics that need to be modeled at this level are data, functional, and non-functional semantics.
4. The middleware services tier must be able to provide services that will allow the process requirements for dynamism and adaptation to be fulfilled. The middleware services tier must be aware of the data semantics to provide service selection and data mediation services. The functional semantics must be explicated at this level in order to facilitate service selection. Further the middleware services tier needs to be aware of non-functional semantics to do process configuration. An example of a configuration service would be the multi-paradigm constraint analysis discussed in (Aggarwal et al., 2004) and (Verma, 2006). The middleware services tier is also responsible for event identification, subscription and adaptation. To realize these services, execution semantics must be captured at this level.

Figure 1 illustrates the four-tier approach to model, enact and execute business processes. In the next section, we briefly define semantic templates (introduced in (Verma, 2006) and (Gomadam et al., 2007)) and discuss an approach to realize the 4×4 Model using semantic templates.

4 Realizing the 4×4 Model using Semantic Templates

A *semantic template* is a service interface specification along with the service- and operation-level policies, which are enhanced with semantic metadata from functional and non-functional ontologies. The semantic metadata capture the data, functional and non-functional semantics of the Web service and is expressed using model references. A conceptual model of a semantic template is presented in figure 2. One way of implementing a semantic template is by using the SAWSDL (SAWSDL, 2007) specification to annotate the standard WSDL (WSDL, 2001) documents for modeling data and functional semantics. Non-functional semantics can be modeled by attaching semantically annotated assertions. In the context

of REST-based services, adding annotations to URIs and XML types using microformats creates semantic templates.

As illustrated in figure 2, a semantic template models the data, functional and non-functional semantics. The semantic template as illustrated consists of an operation, input and output elements along with their model references. The data and the functional semantics are captured by these elements. The non-functional semantics is captured by the term policy element. Each term policy is a collection of assertions that model a constraint. To each constraint, a model reference grounding the constraining variable to a concept in a non-functional ontology is attached. If the term policy element is not attached to any operation element, it captures the template level policy.

Although this model is sufficient to capture the semantics at the level of process modeling tier and partner services tier, the lack of execution semantics in the semantic template model makes it a partial model for the Workflow Enactment Tier and the middleware services tier. To overcome this limitation, we propose the following approach for capturing execution semantics. In an earlier paper (Krishnakumar and Sheth, 1995), we proposed a task skeleton based model for capturing the various execution states of a task in a workflow. The state-based task skeleton model captured a set of externally visible task states, including a start state and a set of termination states. The events that can trigger state transitions are also captured. In order to capture the execution semantics, we use the task skeleton model to capture the various execution states. The events are identified using the approach proposed in (Gomadam et al., 2007) and the transitions between the states are modeled. This task skeleton model is attached to a semantic template to create a complete model of data, functional, non-functional, and execution semantics.

4.1 Capturing the semantics at different tiers using semantic templates

We will now demonstrate the modeling of semantics in the different tiers using semantic templates.

1. In the Business Process Tier, the semantic template capture the functional and the non-functional semantics. The operation element in the semantic template, along with the model reference attribute, captures the functional semantics of the business process. The non-functional semantics are captured using the assertions and the assertion model reference in the template terms. Here the semantic template captures the functional and non-functional semantics that are rel-

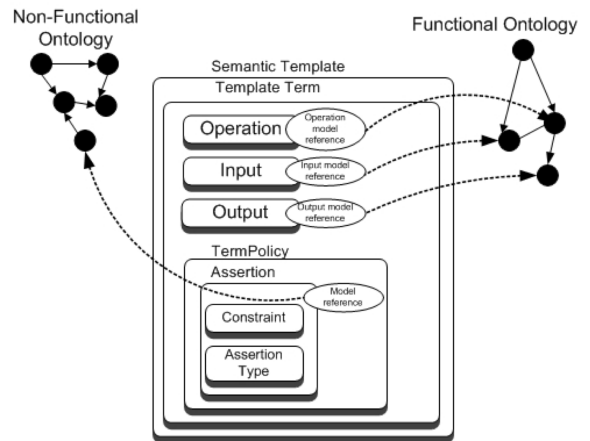


Figure 2: Conceptual Model of a Semantic Template

evant at this tier. In case of conventional SOAP based SOA implementations, the semantic template would model the semantics using SAWSDL and enhanced policy constructs. For REST-based lightweight processes, the templates would model the semantics on an annotated XML document with the model references captured using microformats such as GRDDL (GRDDL, 2007) and RDFa (RDFa, 2007).

2. In the Workflow Enactment Tier, the semantic template models the data semantics using the input and output elements, the functional semantics using the operation element and the non-functional semantics using the assertions and term policy elements. Further the semantic template allows for modeling template-level policy and operation-level policy. This is described in (Gomadam et al., 2007). This flexibility allows for separation of non-functional requirements at the level of a partner and at the level of partner operations. Task skeletons are attached to the semantic template by way of additional assertions in the template term. The events for the transition are identified from the process-level requirement and are modeled into the assertions that capture the task skeleton. For the SOAP driven SOA implementations the workflow enactment can be done using BPEL. For REST-based lightweight enactments or smashups, one approach to process composition is by using java script alongside RDFa enhanced JSON for capturing the types.
3. At the Partner Services Tier, semantic templates can capture the data, functional and non-functional semantics in the same way as in the workflow enactment tier. If the service provider offers REST-based services, then the syntactic

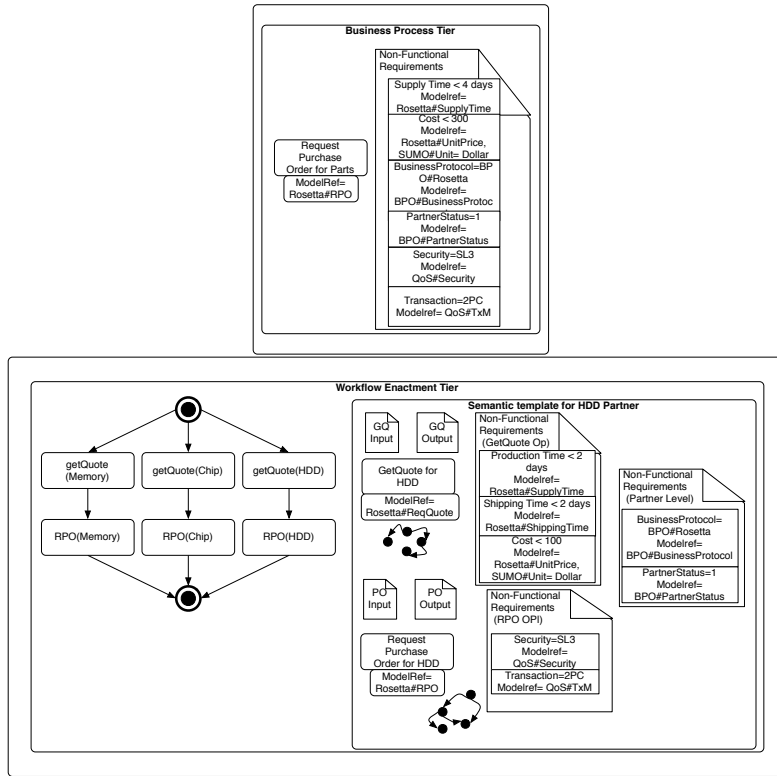


Figure 3: Using Semantic Template to model the Business Process and Workflow Enactment Tiers

sugar of the template would be in SA-REST and not SAWSDL.

4. Wohlstadter et al. (Wohlstadter et al., 2006) present an approach to creating service-oriented middleware systems. We extend that notion to capture the data, functional, and non-functional attributes of such middleware using SAWSDL. Extending the model proposed in (Wohlstadter et al., 2006), we define the operations for partner selection, process configuration, partner invocation, and process adaptation as middleware service operations. The semantic templates capture the semantics associated with each of these operations.

- (a) For the partner selection operation, the semantic template capture the domain models the middleware is aware of along with the semantics of the logical formalisms used in modeling the preconditions and effects.
- (b) The process configuration operation is enhanced with the capabilities of the system to support policy languages and domain models.
- (c) The semantics of data mediation are captured in the partner invocation operation.

- (d) The formal modeling of various exceptions and the adaptation rules for the same are captured in the process adaptation operation

5 The 4×4 Model in action during modeling, enactment and execution

In this section we demonstrate the use of the 4×4 Model using the scenario described in 2, to capture the process- and partner-level constraints. We further illustrate an enactment and execution scenario based on our model. Modeling at the level of the business process tier consists of capturing the functional requirement, which is to create a purchase order for various parts used in a gaming hardware. The non-functional requirements to be captured are the constraints on the cost, supply time, business protocol to be followed by the partners, and relationship of the supplier to the manufacturer. The semantic template to capture these constraints is illustrated in figure 3.

In figure 3, the manufacturer captures the functional semantics of the purchase order operation using

the operation construct of the semantic template. The non-functional constraints include the constraint on cost ($\text{unitPrice} < 300$) and supply time ($\text{Supply time} \leq 4$ days), in addition to the constraint that the supplier must follow the RosettaNet business protocol. The manufacturer would also like to order parts only from preferred suppliers. Figure 3 illustrates the modeling of process-level constraints using semantic templates. This process-level constraint is broken down into partner-level constraints. Figure 3 illustrates the semantic template model for capturing the semantics for the hard drive supplier.

In figure 3 we illustrate a model that captures the requirements for the hard drive supplier. This modeling is done at the workflow enactment tier. In this model we break down the task of creating a purchase order into two tasks: (1) getting a quote from the supplier and (2) creating a purchase order. It must be noted here that in our approach as outlined in (Verma et al., 2006), we focus on dynamic configuration as opposed to composition. Hence this breakdown of operation is done manually. However in our more recent work (Wu et al., 2007), we attempt to use planning based techniques to automatically create this breakdown. The functional semantics is captured by annotating the template operations, `getQuote_GamingHDD` and `PO_Order_GamingHDD`, with concepts from the RosettaNet Ontology (Lsdis Lab, 2004). The constraints are modeled at the level of the partner operations. The `getQuote` operation has constraints on cost and supply time. The `PO_Order` operation has constraints on security level and transactions. The partner status and the partner protocol are captured as template-level constraints. This completes the functional and non-functional modeling. Using task skeleton for the `getQuote` and `Purchase-Order` operations captures execution semantics. We will now illustrate an enactment and execution of business processes using semantic templates at the business process tier and the semantic template modeling of partners at the middleware enactment tier. Figure 4 illustrates the interaction between the various tiers during enactment and execution.

The manufacturer captures the semantics at the business process tier, as described earlier in this section. The semantic templates for the partners are also captured as described earlier. A BPEL-based workflow is enacted using the semantic templates as the process partners. This workflow is deployed in the middleware that provides service selection, process configuration, and adaptation services. The workflow enactment during execution utilizes these different services and the partners are bound to the process. This approach, also referred to as dynamic binding, is

described in (Verma et al., 2006). If the process composition were to use REST-based SOA implementations, then the compositions (also called smashups) are realized using javascript and JSON. The templates are described using SA-REST.

6 Conclusions

This paper presented a comprehensive way to use semantics to link the different tiers of a process and service architecture. Semantic Templates provide a unifying construct in our 4×4 Semantic Model that encompasses the four tiers and uses four types of semantics. Our hope is that this broad and comprehensive model becomes a basis of studying a number of exciting research efforts being carried out in the Business Process Management, Semantic Web Services, Services Oriented Architecture, Middleware and Distributed Computing communities.

REFERENCES

- Aggarwal, R., Verma, K., Miller, J. A., and Milnor, W. (2004). Constraint driven web service composition in meteor-s. In *IEEE SCC*, pages 23–30.
- Akkiraju, R., Farrell, J., Miller, J., Nagarajan, M., Schmidt, M., Sheth, A. P., and Verma, K. (2004). Web Service Semantics – WSDL-S. Technical report, LSDIS Lab and IBM Corporation.
- Cardoso, J., Sheth, A. P., Miller, J. A., Arnold, J., and Kochut, K. (2004). Quality of service for workflows and web service processes. *J. Web Sem.*, 1(3):281–308.
- Gomadam, K., Ranabahu, A., Sheth, A. P., and Verma, K. (2007). A Semantic Framework for Identifying Events in SOA. Technical report, kno.e.sis center, Wright State University.
- GRDDL (2007). <http://www.w3.org/2004/01/rdxh/spec>.
- Krishnakumar, N. and Sheth, A. P. (1995). Managing heterogeneous multi-system tasks to support enterprise-wide operations. *Distributed and Parallel Databases*, 3(2):155–186.
- Lsdis Lab (2004). Rosettanet Ontology, <http://lsdis.cs.uga.edu/projects/meteor-s/wsdls/ontologies/rosetta.owl>.
- Microsoft Corporation (2006). Moving to BizTalk Server 2006, Microsoft White Paper.
- Oldham, N., Verma, K., Sheth, A. P., and Hakimpour, F. (2006). Semantic ws-agreement partner selection. In *WWW '06: Proceedings of the 15th*

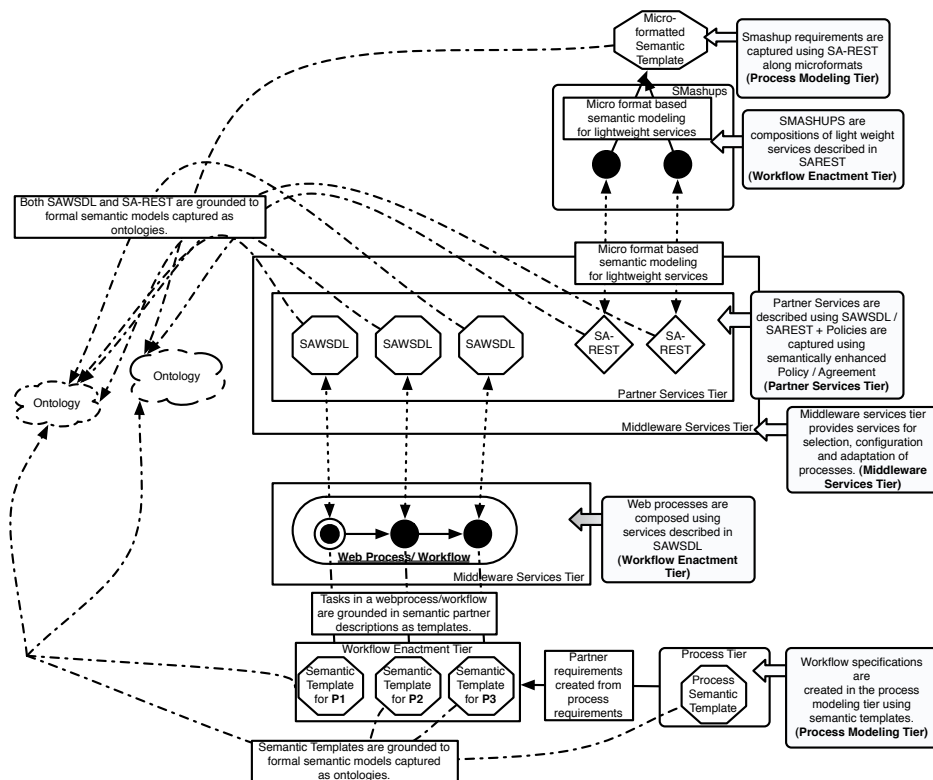


Figure 4: Using Semantic Template to model the Business Process and Workflow Enactment Tiers

international conference on World Wide Web, pages 697–706, New York, NY, USA. ACM Press.

RDFa (2007). <http://www.w3.org/tr/xhtml-rdfa-primer/>.

SAWSDL (2007). Semantic Annotations for WSDL and XML Schema. Technical report, W3 Consortium.

Sheffi, Y. (2005). *The Resilient Enterprise: Overcoming Vulnerability for Competitive Advantage*. The MIT Press.

Sheth, A. P. (2003). Semantic Web Process Lifecycle: Role of Semantics in Annotation, Discovery, Composition and Orchestration. WWW 2003 Workshop on E-Services and the Semantic Web, Budapest, Hungary.

Sivashanmugam, K., Verma, K., Sheth, A. P., and Miller, J. A. (2003). Adding semantics to web services standards. In *ICWS*, pages 395–401.

Verma, K. (2006). *Configuration And Adaptation of Semantic Web Processes*. PhD thesis, University of Georgia.

Verma, K., Gomadam, K., Lathem, J., Sheth, A. P.,

and Miller, J. (2006). Semantics Enabled Dynamic Process Configuration. Technical report, LSDIS Lab.

Wohlstadter, E., Tai, S., Mikalsen, T. A., Diament, J., and Rouvellou, I. (2006). A Service-oriented Middleware for Runtime Web Services Interoperability. In *ICWS*, pages 393–400.

WSDL (2001). Web Services Description Language (WSDL) 1.1.

Wu, Z., Ranabahu, A., Sheth, A. P., and Miller, J. (2007). Automatic Composition of Semantic Web Services using Process and Data Mediation. Technical report, kno.e.sis center, Wright State University.